What is Claimed:

1. A method of clearing obsolete entries from a first one of a plurality of mapping caches, each of the plurality of mapping caches being associated with a corresponding one of a plurality of processing units of a computing device, each of the caches being used to translate virtual addresses to physical addresses and storing mappings based on an address translation map, the method comprising:

maintaining a counter;

updating said counter each time the first one of the plurality of mapping caches is flushed;

recording said counter's value in response to a change in the address translation map;

determining, based on a comparison of said counter's value with the recorded counter value, that the first one of the plurality of mapping caches has not definitely been flushed since said change in the address translation map occurred; and

flushing the first one of the plurality of mapping caches.

2. The method of claim 1, wherein said counter is one of a plurality of counters, each counter being associated with a corresponding one of the mapping caches.

3. The method of claim 2, wherein each of the plurality of counters is updated when a counter's corresponding mapping cache is flushed.

4. The method of claim 1, wherein the address translation map comprises links to pages of memory including a first page, and wherein said change comprises placing the address translation map in a state in which the address translation map does not contain any links to said first page.

5. The method of claim 1, wherein the address translation map defines portions of a memory that are readable by an entity, said portions of said memory including a first portion, and wherein said change comprises placing the address translation map in a state in which said first portion is not readable by said entity.

6. The method of claim 1, wherein the address translation map defines portions of a memory that are writeable by an entity, said portions of said memory including a first portion, and wherein said change comprises placing the address translation map in a state in which said first portion is not writeable by said entity.

7. The method of claim 1, wherein a policy defines permissible access to a memory, and wherein the method further comprises:

controlling the content of the address translation map such that the address translation map does not expose to an entity virtual address mappings that would permit said entity to access said memory in violation of said policy;

wherein said change comprises either a modification to the map that places or maintains the map in conformance with said policy, or a modification to the map that limits said entity's write access to the map.

8. The method of claim 7, wherein the address translation map comprises a link to a portion of said memory, wherein controlling the content of the address translation map comprises making said portion of said memory inaccessible to said entity, and wherein said change comprises removing all links to said portion of said memory from the address translation map.

9. The method of claim 1, further comprising:

determining that an event has arisen that potentially makes use of the state of the address translation map prior to said change;

wherein flushing the first one of the plurality of map caches is performed in response to the determination that said event has arisen.

10. The method of claim 9, wherein said event comprises a translation of a virtual address to a physical address that is contained in a portion of memory that was de-linked from the address translation map by said change.

11. A system for managing the use of address mapping caches, the system comprising:

a plurality of processors, each of the processors having a mapping cache and a counter associated therewith;

a memory that stores an address translation map, each of the mapping caches storing mappings based on said address translation map, there being a policy that governs access to said memory, the contents of the address translation map being controlled to prevent exposure of mappings that permit access to said memory in violation of said policy;

first logic that flushes a first one of the mapping caches and that increments a first one of the counters when said first one of the mapping caches is flushed;

second logic that records the current value of said first counter in response to a change in said address translation map or in a property regarding said address translation map, the recorded counter value being stored in association with said change;

third logic that compares the recorded counter value with the current value of the first counter and that causes said first one of the mapping caches to be flushed if the comparison indicates that said first one of the mapping caches has not been flushed since the change.

12. The system of claim 11, wherein each of the processors is associated with said first one of the counters, and wherein said first logic increments said first one of the counters when any of the mapping caches is flushed.

13. The system of claim 11, wherein each of the mapping caches is associated with a different one of a plurality of counters, and wherein said first logic increments the counter corresponding to a given mapping cache when the given mapping cache is flushed.

14. The system of claim 11, wherein said change comprises placing said address translation map in a state in which all links to a first page of said memory are removed from said address translation map.

15. The system of claim 11, wherein said change comprises placing said address translation map in a state in which there is no writeable link to a first page of said memory.

16. The system of claim 11, wherein said third logic is invoked in response to a detection that a result of said change is to be used.

17. The system of claim 16, wherein said change comprises placing said address translation map in a state in which all links to a first page of said memory are removed from said address translation map, and wherein said detection is based on a virtual address having been translated to a location on said first page.

18. A computer-readable medium having encoded thereon computer-executable instructions to perform a method of managing the flushing of a translation lookaside buffer that caches address mappings, the method comprising:

receiving an access request that indicates a target location by virtual address;

translating said virtual address to obtain a physical address of said target location;

comparing (1) a stored counter value associated with a page that comprises said target address with (2) a current counter value;

determining based on the comparison between said stored counter value and said current counter value that the translation lookaside buffer has not been flushed since an event affecting a mapping of said page has been modified; and

flushing the translation lookaside buffer.

19. The computer-readable medium of claim 18, wherein the address mappings are defined by an address translation map stored in a memory, the translation lookaside buffer caching mappings that are based on said address translation map.

20. The computer-readable medium of claim 18, wherein a policy defines accessibility of a memory to a software entity, and wherein said event comprises removing links to a page of said memory from an address translation map on which said address mappings are based, said page being inaccessible to said software entity under said policy.

21. The computer-readable medium of claim 18, wherein a policy defines accessibility of a memory to a software entity, and wherein said event comprises adjusting an address translation map to make mappings to a page non-writeable, wherein either: (1) said policy defines said page as being non-writeable by said software entity, or (2) said page stores a portion of said address translation map.

22. The computer-readable medium of claim 18, wherein an address translation control mechanism determines the membership of a set of pages of a memory that may be used to store portions of an address translation map, and wherein said event comprises a change in membership of said set.

23. A method of clearing obsolete entries from a first one of a plurality of mapping caches, each of the plurality of mapping caches being associated with a corresponding one of a plurality of processing units of a computing device, each of the caches being used to translate virtual addresses to physical addresses and storing mappings based on an address translation map, the method comprising:

maintaining an object based on which the sequential progression of events can be determined;

recording said object's value in response to a change to the address translation map that removes one or more of, (1) read access, or (2) read/write access, from a page;

in response to a request to access said page, determining based on a comparison of said object's current value with the recorded value that the first one of the plurality of mapping caches has been flushed since said change occurred; and

if it cannot be determined with certainty based on said comparison that the first one of the plurality of mapping caches has been flushed since said change occurred, then flushing the first one of the plurality of mapping caches.

24. The method of claim 23, wherein said object comprises a clock.

25. The method of claim 23, wherein said object comprises a counter that is incremented each time the first one of the plurality of mapping caches is flushed.

26. The method of claim 23, wherein said object comprises a plurality of counters, each of the counters corresponding to one of the plurality of mapping caches, wherein each of the counters is incremented when the counter's corresponding mapping cache is flushed.

27. The method of claim 23, wherein said change comprises removing read access to said page, and wherein said access request comprises a request to read said page.

28. The method of claim 23, wherein said change comprises removing read/write access to said page, and wherein said access request comprises a request to write said page.